



Docket Number Y0998-331

## REDUCED-ERROR PROCESSING OF TRANSFORMED DIGITAL DATA

### 2 CROSS REFERENCES

3       The present application is related to the following  
4 applications even dated herewith: Attorney docket number  
5 Y0998-372, entitled, "Transform-domain correction of real-  
6 domain errors," by inventors J. Mitchell et al., and  
7 Attorney docket number Y0998-373, entitled, "Error reduction  
8 in transformed digital data," by inventors M. Bright et al.,  
9 which are incorporated herein in entirety by reference.

### 10 FIELD OF THE INVENTION

11       This invention relates to transform coding of digital  
12 data, specifically to real domain processing of transform  
13 data. More particularly, this invention relates to  
14 reduced-error digital processing of inverse transformed  
15 data.

### 16 BACKGROUND OF THE INVENTION

17       Transform coding is the name given to a wide family of  
18 techniques for data coding, in which each block of data to  
19 be coded is transformed by some mathematical function prior  
20 to further processing. A block of data may be a part of a  
21 data object being coded, or may be the entire object. The

1 data generally represent some phenomenon, which may be for  
2 example a spectral or spectrum analysis, an image, an audio  
3 clip, a video clip, etc. The transform function is usually  
4 chosen to reflect some quality of the phenomenon being  
5 coded; for example, in coding of audio, still images and  
6 motion pictures, the Fourier transform or Discrete Cosine  
7 Transform (DCT) can be used to analyze the data into  
8 frequency terms or coefficients. Given the phenomenon being  
9 coded, there is generally a concentration of the information  
10 into a few frequency coefficients. Therefore, the  
11 transformed data can often be more economically encoded or  
12 compressed than the original data. This means that  
13 transform coding can be used to compress certain types of  
14 data to minimize storage space or transmission time over a  
15 communication link.

16 An example of transform coding in use is found in the  
17 Joint Photographic Experts Group (JPEG) international  
18 standard for still image compression, as defined by *ITU-T*  
19 *Rec. T.81 (1992) ISO/IEC 10918-1:1994, Information*  
20 *technology — Digital compression and coding of*  
21 *continuous-tone still images, Part 1: Requirements and*  
22 *Guidelines*. Another example is the Moving Pictures Experts  
23 Group (MPEG) international standard for motion picture  
24 compression, defined by *ISO/IEC 11172:1993, Information*  
25 *Technology — Coding of moving pictures and associated audio*  
26 *for digital storage media at up to about 1,5 Mbits/s*. This  
27 MPEG-1 standard defines systems for both video compression  
28 (Part 2 of the standard) and audio compression (Part 3). A  
29 more recent MPEG video standard (MPEG-2) is defined by *ITU-T*  
30 *Rec. H.262 ISO/IEC 13818-2: 1996 Information Technology —*

1    *Generic Coding of moving pictures and associated audio --*  
2    *Part 2: video.* A newer audio standard is *ISO/IEC 13818-3:*  
3    *1996 Information Technology -- Generic Coding of moving*  
4    *pictures and associated audio -- Part 3: audio.* All three  
5    image international data compression standards use the DCT  
6    on 8x8 blocks of samples to achieve image compression. DCT  
7    compression of images is used herein to give illustrations  
8    of the general concepts put forward below; a complete  
9    explanation can be found in Chapter 4 "The Discrete Cosine  
10    Transform (DCT)" in W. B. Pennebaker and J. L. Mitchell,  
11    *JPEG: Still Image Data Compression Standard*, Van Nostrand  
12    Reinhold: New York, (1993).

13        Wavelet coding is another form of transform coding.  
14    Special localized basis functions allow wavelet coding to  
15    preserve edges and small details. For compression the  
16    transformed data is usually quantized. Wavelet coding is  
17    used for fingerprint identification by the FBI. Wavelet  
18    coding is a subset of the more general subband coding  
19    technique. Subband coding uses filter banks to decompose the  
20    data into particular bands. Compression is achieved by  
21    quantizing the lower frequency bands more finely than the  
22    higher frequency bands while sampling the lower frequency  
23    bands more coarsely than the higher frequency bands. A  
24    summary of wavelet, DCT, and other transform coding is given  
25    in Chapter 5 "Compression Algorithms for Diffuse Data" in  
26    Roy Hoffman, *Data Compression in Digital Systems*, Chapman  
27    and Hall: New York, (1997).

28        In any technology and for any phenomenon represented by  
29    digital data, the data before a transformation is performed  
30    are referred to as being "in the real domain". After a

1 transformation is performed, the new data are often called  
 2 "transform data" or "transform coefficients", and referred  
 3 to as being "in the transform domain". The function used to  
 4 take data from the real domain to the transform domain is  
 5 called the "forward transform". The mathematical inverse of  
 6 the forward transform, which takes data from the transform  
 7 domain to the real domain, is called the respective "inverse  
 8 transform".

9 In general, the forward transform will produce  
 10 real-valued data, not necessarily integers. To achieve data  
 11 compression, the transform coefficients are converted to  
 12 integers by the process of quantization. Suppose that  $(\lambda_i)$   
 13 is a set of real-valued transform coefficients resulting  
 14 from the forward transform of one unit of data. Note that  
 15 one unit of data may be a one-dimensional or two-dimensional  
 16 block of data samples or even the entire data. The  
 17 "quantization values"  $(q_i)$  are parameters to the encoding  
 18 process. The "quantized transform coefficients" or  
 19 "transform-coded data" are the sequence of values  $(a_i)$   
 20 defined by the quantization function  $Q$ :

$$21 \quad a_i = Q(\lambda_i) = \left\lfloor \frac{\lambda_i}{q_i} + 0.5 \right\rfloor \quad (1)$$

22 where  $\lfloor x \rfloor$  means the greatest integer less than or equal to  $x$ .

23 The resulting integers are then passed on for possible  
 24 further encoding or compression before being stored or  
 25 transmitted. To decode the data, the quantized coefficients  
 26 are multiplied by the quantization values to give new  
 27 "dequantized coefficients"  $(\lambda'_i)$  given by

$$28 \quad \lambda'_i = q_i a_i. \quad (2)$$

1       The process of quantization followed by dequantization  
2 (also called inverse quantization) can thus be described as  
3 "rounding to the nearest multiple of  $q_i$ ". The quantization  
4 values are chosen so that the loss of information in the  
5 quantization step is within some specified bound. For  
6 example, for audio or image data, one quantization level is  
7 usually the smallest change in data that can be perceived.  
8 It is quantization that allows transform coding to achieve  
9 good data compression ratios. A good choice of transform  
10 allows quantization values to be chosen which will  
11 significantly cut down the amount of data to be encoded.  
12 For example, the DCT is chosen for image compression because  
13 the frequency components which result produce almost  
14 independent responses from the human visual system. This  
15 means that the coefficients relating to those components to  
16 which the visual system is less sensitive, namely the  
17 high-frequency components, may be quantized using large  
18 quantization values without perceptible loss of image  
19 quality. Coefficients relating to components to which the  
20 visual system is more sensitive, namely the low-frequency  
21 components, are quantized using smaller quantization values.

22       The inverse transform also generally produces  
23 non-integer data. Usually the decoded data are required to  
24 be in integer form. For example, systems for the playback  
25 of audio data or the display of image data generally accept  
26 input in the form of integers. For this reason, a transform  
27 decoder generally includes a step that converts the  
28 non-integer data from the inverse transform to integer data,  
29 either by truncation or by rounding to the nearest integer.  
30 There is also often a limit on the range of the integer data

1 output from the decoding process in order that the data may  
2 be stored in a given number of bits. For this reason the  
3 decoder also often includes a "clipping" stage that ensures  
4 that the output data are in an acceptable range. If the  
5 acceptable range is  $[a,b]$ , then all values less than  $a$  are  
6 changed to  $a$ , and all values greater than  $b$  are changed to  
7  $b$ .

8       These rounding and clipping processes are often  
9 considered an integral part of the decoder, and it is these  
10 which are the cause of inaccuracies in decoded data and in  
11 particular when decoded data are re-encoded. For example,  
12 the JPEG standard (Part 1) specifies that a source image  
13 sample is defined as an integer with precision  $P$  bits, with  
14 any value in the range 0 to  $2^P - 1$ . The decoder is  
15 expected to reconstruct the output from the inverse discrete  
16 cosine transform (IDCT) to the specified precision. For the  
17 baseline JPEG coding  $P$  is defined to be 8; for other  
18 DCT-based coding  $P$  can be 8 or 12. The MPEG-2 video  
19 standard states in Annex A (Discrete cosine transform) "The  
20 input to the forward transform and the output from the  
21 inverse transform is represented with 9 bits."

22       For JPEG the compliance test data for the encoder  
23 source image test data and the decoder reference test data  
24 are 8 bit/sample integers. Even though rounding to integers  
25 is typical, some programming languages convert from floating  
26 point to integers by truncation. Implementations in  
27 software that accept this conversion to integers by  
28 truncation introduce larger errors into the real-domain  
29 integer output from the inverse transform.

1       The term "high-precision" is used herein to refer to  
2 numerical values which are stored to a precision more  
3 accurate than the precision used when storing the values as  
4 integers. Examples of high-precision numbers are  
5 floating-point or fixed-point representations of numbers.

#### 6       SUMMARY OF THE INVENTION

7       In light of the problems described above regarding  
8 inaccuracies caused by digital processing techniques and by  
9 such things as rounding and clipping after the inverse  
10 transform of transform data, one aspect of this invention  
11 provides a method for processing transform data in the real  
12 domain. This method reduces the undesired errors in the  
13 data produced by such things as rounding to integers and  
14 clipping to an allowed range after the inverse transform.  
15 In an embodiment, this method includes: performing the  
16 inverse transform of the transform data such that the  
17 real-domain data produced are in the form of high-precision  
18 numbers; processing these high-precision numbers; and  
19 converting the processed high-precision numbers to integers  
20 and clipping to an allowed range only after the processing  
21 stage is complete.

22       It is another aspect of this invention to provide a  
23 method for processing transform-coded data in the real  
24 domain which reduces the undesired errors in the data  
25 produced by the converting to integers and clipping to an  
26 allowed range after the inverse transform. In an  
27 embodiment, the method includes: performing the inverse

1 quantization of the transform-coded data; performing the  
2 inverse transform of the transform data thus produced, such  
3 that the real-domain data produced are in the form of  
4 high-precision numbers; processing these high-precision  
5 numbers; and converting the processed high-precision numbers  
6 to integers and clipping to an allowed range only after the  
7 processing stage is complete.

8       Still another aspect of the present invention is to  
9 provide a method for processing transform-coded data in the  
10 real domain to produce new transform-coded data, which  
11 reduces the error produced by converting to integers and  
12 clipping to an allowed range after the inverse transform.  
13 In an embodiment, this method includes: performing the  
14 inverse quantization of the transform-coded data; performing  
15 the inverse transform of the transform data thus produced,  
16 such that the real-domain data produced are in the form of  
17 high-precision numbers; processing these high-precision  
18 numbers; performing the forward transform on the processed  
19 high-precision numbers; and performing quantization on the  
20 new transform data. If the errors in the forward and  
21 inverse transforms and in the processing are sufficiently  
22 small, there will be no undesirable errors produced in the  
23 new quantized transform-domain data.

24       There is no requirement that the input data to the  
25 methods described herein need come from a single data  
26 source. Thus, this invention is not restricted to the  
27 real-domain processing of data from a single source, but  
28 also applies to real-domain processing of data from multiple  
29 sources, such as the merging of images or audio data.



1       The quantization described in the background is the  
2 linear quantization used in international image data  
3 compression standards such as JPEG and MPEG. There is no  
4 requirement that the quantization be linear. Any mapping  
5 that reduces the number of transform data levels in a  
6 deterministic way can be used with this invention. The  
7 quantization step has been described mathematically with a  
8 division in Equation (1). Actual embodiments may use a  
9 lookup table or a sequence of comparisons to achieve similar  
10 results.

11       It is a further aspect of the invention to provide  
12 apparatus, a computer product and an article of manufacture  
13 comprising a computer usable medium having computer readable  
14 program code means embodied therein for causing a computer  
15 to perform the methods of the present invention.

#### 16   BRIEF DESCRIPTION OF FIGURES

17       These and other objects, features, and advantages of  
18 the present invention will become apparent upon further  
19 consideration of the following detailed description of the  
20 invention when read in conjunction with the drawing figures,  
21 in which:

22       FIG. 1(a) is a block diagram showing a method for  
23 performing an inverse transform;

24       FIG. 1(b) is a block diagram showing a system for  
25 performing an inverse transform;

26       FIG. 2(a) is a block diagram showing a method for  
27 decoding transform-coded data;

1        FIG. 2(b) is a block diagram showing a system for  
2 decoding transform-coded data;

3        FIG. 3 is a block diagram showing a method for the  
4 real-domain processing of transform data;

5        FIG. 4 is a block diagram showing a method for  
6 performing an inverse transform followed by a forward  
7 transform, and demonstrating the multi-generation problem;

8        FIG. 5 is a block diagram showing a method for decoding  
9 and re-encoding transform-coded data, and demonstrating the  
10 multi-generation problem;

11       FIG. 6 is a block diagram showing a method for  
12 performing an inverse transform, real-domain data  
13 manipulation and a forward transform, and demonstrating the  
14 multi-generation problem;

15       FIG. 7(a) is a block diagram showing a method for  
16 performing real-domain processing of JPEG DCT-coded image  
17 data, which exhibits the multi-generation problem;

18       FIG. 7(b) is a block diagram showing a system for  
19 performing real-domain processing of JPEG DCT-coded image  
20 data, which exhibits the multi-generation problem;

21       Fig. 8(a) gives the JPEG example luminance quantization  
22 matrix;

23       Fig. 8(b) gives the JPEG example chrominance  
24 quantization matrix;

25       FIG. 8(c) is a numerical example of how real-domain  
26 rounding can cause significant errors in 8x8 block DCT coded  
27 data;

1        FIG. 8(d) is a numerical example of how real-domain  
2 truncation can cause significant errors in 8x8 block DCT  
3 coded data;

4        FIG. 8(e) is a series of graphs illustrating how  
5 real-domain clipping can cause errors in one-dimensional  
6 discrete cosine transform-coded data;

7        FIG. 8(f) and FIG. 8(g) are a numerical example of how  
8 real-domain clipping can cause significant errors in 8x8  
9 block DCT coded data;

10       FIG. 9 is a block diagram showing a method performing  
11 multiple iterations of the process described in FIG. 5, and  
12 exhibiting the multi-generation problem;

13       FIG. 10 is a block diagram showing a method for  
14 performing multiple iterations of real-domain manipulations,  
15 and exhibiting the multi-generation problem;

16       FIG. 11(a) is a block diagram showing an example of a  
17 method for reduced-error processing of transform data in  
18 accordance with the present invention;

19       FIG. 11(b) is a block diagram showing an example of a  
20 system for reduced-error processing of transform data in  
21 accordance with the present invention;

22       FIG. 12(a) is a block diagram showing an example of a  
23 method for performing an inverse transform followed by a  
24 forward transform, such that this process is lossless in  
25 accordance with the present invention;

26       FIG. 12(b) is a block diagram showing an example of a  
27 system for performing an inverse transform followed by a

1 forward transform, such that this process is lossless in  
2 accordance with the present invention;

3 FIG. 13(a) is a block diagram showing an example of a  
4 method for performing real-domain manipulation of transform  
5 data with reduced error followed by a forward transform in  
6 accordance with the present invention;

7 FIG. 13(b) is a block diagram showing an example of a  
8 system for performing real-domain manipulation of transform  
9 data with reduced error followed by a forward transform in  
10 accordance with the present invention;

11 FIG. 14(a) is a block diagram showing an example of a  
12 method for reduced-error processing of transform-coded data  
13 in accordance with the present invention;

14 FIG. 14(b) is a block diagram showing an example of a  
15 system for reduced-error processing of transform-coded data  
16 in accordance with the present invention;

17 FIG. 15(a) is a block diagram showing an example of a  
18 method for decoding and re-encoding transform-coded data  
19 such that this process is lossless in accordance with the  
20 present invention;

21 FIG. 15(b) is a block diagram showing an example of a  
22 system for decoding and re-encoding transform-coded data  
23 such that this process is lossless in accordance with the  
24 present invention;

25 FIG. 16(a) is a block diagram showing an example of a  
26 method for performing real-domain manipulation of  
27 transform-coded data with reduced error in accordance with  
28 the present invention;

1        FIG. 16(b) is a block diagram showing an example of a  
2        system for performing real-domain manipulation of  
3        transform-coded data with reduced error in accordance with  
4        the present invention;

5        FIG. 17(a) is a block diagram showing an example  
6        embodiment of a method for performing real-domain processing  
7        of JPEG-coded image data, such that undesired errors in the  
8        new transform-coded data are reduced or eliminated in  
9        accordance with the present invention;

10       FIG. 17(b) is a block diagram showing an example  
11       embodiment of a system for performing real-domain processing  
12       of JPEG-coded image data, such that undesired errors in the  
13       new transform-coded data are reduced or eliminated in  
14       accordance with the present invention;

15       FIG. 18(a) is a block diagram showing an example of a  
16       method for performing multiple iterations of the real-domain  
17       manipulation of transform-coded data with reduced error,  
18       where each iteration is as described in FIG. 16(a) in  
19       accordance with the present invention;

20       FIG. 18(b) is a block diagram showing an example of a  
21       system for performing multiple iterations of the real-domain  
22       manipulation of transform-coded data with reduced error,  
23       where each iteration is as described in FIG. 16(b) in  
24       accordance with the present invention;

25       FIG. 19(a) shows the same 8x8 block numerical starting  
26       point of FIG. 8(c) using the high-precision numbers as input  
27       to the forward transform instead of the rounded numbers;

1        FIG. 19(b) shows the same 8x8 block numerical starting  
2 point of FIG. 8(d) using the high-precision numbers as input  
3 to the forward transform instead of the truncated numbers;

4        FIG. 19(c) shows the same 8x8 block numerical steps as  
5 FIG. 8(f); and

6        FIG. 19(d) shows the numerical results when the output  
7 of the inverse DCT with rounding, but before clipping, is  
8 input to the forward transform followed by quantization.

#### 9        DESCRIPTION OF THE PROBLEM

10       This invention provides methods, systems, and computer  
11 products which reduce or eliminate errors introduced by the  
12 processing of digital data. Firstly, the source of the  
13 error is analyzed and described. This is followed by a  
14 presentation of the invention concepts for error reduction  
15 and elimination. It is particularly noted that data  
16 manipulation and/or processing as employed here-to-before  
17 used digital techniques contaminated by the continued  
18 introducing of errors by the respective implementation of  
19 digital processing. These techniques employed for years are  
20 responsible for an inability to maintain original data  
21 precision and the continued deterioration of the data  
22 representing the phenomenon as more processing is performed.  
23 This is particularly detrimental when a process is performed  
24 on data which contain errors imparted on the data by  
25 previous processes. This results in the continued  
26 impairment of the data which thereby becomes less and less  
27 useful as more and more processes are performed thereupon.

1       The seriousness of the problem as realized by the  
2 inventors of the present invention is described forthwith.  
3 It is noted that in the figures presented herein, optional  
4 steps are often shown with dashed lines and/or boxes.

5       It is noted that the concepts of the present invention  
6 are useful in almost any digital processing technology.  
7 However, the subsequent description is mostly related to  
8 image data. This is because of the general availability and  
9 continued usage of image data compression standards which  
10 are employed worldwide. These standards require the  
11 introduction into the digital data of the errors to be  
12 described and the continued employment and processing of the  
13 error contaminated data. These standards basically teach  
14 away from the present invention. Thus image technology is a  
15 good example for describing the present invention.

16       Figure 1(a) shows an inverse transform method 100.  
17 Transform-domain data 'A' 110 are acted on by the inverse  
18 transform 120, which produces high-precision real-valued  
19 data 130. The high-precision data 130 are converted to  
20 integers and clipped 140 to produce integer real-domain data  
21 150. In some cases, the integer-valued data are optionally  
22 sent to an output device 160.

23       Figure 1(b) shows an inverse transform system 105.  
24 Transform-domain data 'A' 115 are acted on by the inverse  
25 transformer 125, which produces high-precision real-valued  
26 data 135. The high-precision data 135 are input to the  
27 integer converter and clipper 145 to produce integer  
28 real-domain data 155. In some cases, the integer-valued

1 data are optionally input to an output device 165 such as a  
2 display monitor, a television set, or an audio player.

3 Figure 2(a) shows a method 200 for decoding  
4 transform-coded (i.e. quantized) data. The integer  
5 transform-coded data 'B' 210 are inverse quantized 220 (i.e.  
6 dequantized) with quantization values as in Equation (2)  
7 above. The result of the dequantizing step may then be  
8 passed as input to the inverse transform 120, and decoding  
9 proceeds as in Figure 1(a).

10 Figure 2(b) shows a system 205 for decoding  
11 transform-coded (i.e. quantized) data. The integer  
12 transform-coded data 'B' 215 are input to the inverse  
13 quantizer 225 with quantization values as in Equation (2)  
14 above. The result of the dequantizing step is passed as  
15 input to the inverse transformer 125, and decoding proceeds  
16 as in Figure 1(b).

17 One aspect of the present invention is concerned with  
18 the manipulation of both transform data and transform-coded  
19 data. The words "manipulation" and "processing" are used  
20 interchangeably herein. Manipulation may be employed in  
21 order to achieve many different results. For example, image  
22 data must often be processed before printing by scaling  
23 and/or rotation. Data from two sources can be merged as is  
24 performed in chroma-keying of images or mixing of audio  
25 data. Manual manipulation of data is often needed for  
26 editing or color correction. Such manipulation of transform  
27 data are often performed on the integer real-domain data  
28 which results from the transform decoding of Figure 1(a)  
29 and/or Figure 2(a).



1       A process for manipulation of transform data 300 is  
2 shown in Figure 3. Integer data 150 undergo some form of  
3 manipulation 310. If this manipulation 310 does not produce  
4 integer output, the manipulated output 340 is again  
5 converted to integers and clipped 320. The resulting  
6 integer data 330 may be stored, transmitted, and/or  
7 optionally sent to an output device 160. Because the stage  
8 of clipping and converting to integers 140 is performed  
9 before the manipulation which accepts integer input 150, the  
10 resulting errors cause the data output from the manipulation  
11 340 to contain at least small inaccuracies.

12       It is noted that there is no requirement in the data  
13 manipulation processes described above, for the input data  
14 to come entirely from one source. For example, many types  
15 of data manipulation involve the merging of data from two or  
16 more sources. This includes manipulations such as mixing of  
17 audio data or merging of images. The processes illustrated  
18 in the figures and described generally apply equally well to  
19 such types of manipulation. Thus the "input data" used for  
20 any of the processes described may in practice come from  
21 more than one input source.

22       It is often the case that data after manipulation are  
23 to be re-encoded to the transform domain. It is desirable  
24 that the process of decoding and re-encoding, when no  
25 manipulation is performed on the real-domain data, should be  
26 lossless. That is, the data, when the forward transform  
27 operation uses the same transform type operation as the  
28 inverse transform type of transform operation, should result  
29 in exactly the same transform-domain data as was present  
30 initially. However, errors are introduced by the converting

1 to integers and clipping to the allowed range as is  
 2 illustrated in Figure 4. Figure 4 shows the integer data  
 3 150 used as input to the forward transform device 410, which  
 4 accepts integer-valued data as input. The resulting  
 5 transform data 'A1' 420 are different from the original  
 6 transform data 'A' 110 which were the input to the inverse  
 7 transform. This is because the conversion to integers and  
 8 the clipping process 140 have introduced errors into the  
 9 process. The problem caused by the changes in data after  
 10 each iteration, or "generation", of this process is herein  
 11 called the "multi-generation problem".

12 The multi-generation problem is also illustrated for  
 13 transform-coded data in Figure 5. Here the new  
 14 transform-domain data 420 are quantized 510 to produce new  
 15 transform-coded data 'B1' 520. It is important to realize  
 16 that the quantized data can only change if the errors  
 17 produced are larger than half a quantization step:

$$18 \quad Q(\lambda_i + \epsilon) = Q(\lambda_i) \quad \text{if } |\epsilon| < 0.5q_i \quad (3)$$

19 where  $\epsilon$  is the error produced in this transform coefficient.  
 20 This is because each of the  $\lambda_i$  is already a multiple of the  
 21 quantization value, since they have been produced by  
 22 dequantization as in Equation (2). Thus it is advantageous  
 23 to control the errors so that they are sufficiently small.  
 24 When the errors are sufficiently small, the new  
 25 transform-coded data will be exactly the same as the  
 26 original transform-coded data. The maximum possible error  
 27 introduced by the conversion to integers by rounding is half  
 28 the error introduced by truncating during the conversion.

1        Figure 6 shows a case wherein image manipulation is  
2 performed on the data and the resulting modified data are  
3 then re-transformed back to the transform domain. The  
4 integer data 150 are manipulated as was shown in Figure 3 to  
5 produce new integer-valued data 610. These new  
6 integer-valued data 610 are used as the input to the forward  
7 transform 410 to produce new transform data 'A2' 620. The  
8 fact that the process described above without any  
9 manipulation produces changes in the transform data 110  
10 shows that when manipulation is performed there are  
11 undesired changes in the transform data 110 in addition to  
12 those which result from the desired manipulation.

13        An example of a method which embodies the process shown  
14 in Figure 6, is shown in Figure 7(a). The method 700  
15 illustrated performs real-domain manipulation on coded data  
16 such as JPEG-coded image data. The coded data 'C' 710 are  
17 entropy decoded 720, which is defined for JPEG-coded data in  
18 the JPEG standard. The entropy decode step 720 decompresses  
19 the data into quantized DCT coefficients. These quantized  
20 coefficients are inverse quantized 730 and passed to the  
21 inverse transform, which in this system is the  
22 two-dimensional 8x8 inverse DCT 740. The resulting  
23 real-valued image data are rounded to integers and clipped  
24 750 to the allowed range (e.g. [0,255]) to produce  
25 integer-valued image data 754 in the allowed range.

26        If it is necessary to show the data before  
27 manipulation, for example when the image manipulation is an  
28 interactive process, the image can optionally be sent to a  
29 display device 758. The image is then manipulated 762 to

1 produce some desired change. If the result of the  
2 manipulation is non-integer data then the image data may be  
3 converted to integers and clipped to the range e.g. [0,255]  
4 768. In this way the image data 772 may again be displayed  
5 758. The new real-domain image data 772 are passed to the  
6 forward DCT 776 and the resulting DCT coefficients are  
7 quantized 780 to produce new quantized DCT coefficients 784.  
8 These coefficients 784 are then entropy encoded 788 to  
9 produce new coded data 'C1' 792 which are different from the  
10 original coded data 'C' 710. Now the new coded data 'C1'  
11 792 incorporates not only the desired changes made to the  
12 image by the image manipulation 762, but also the errors  
13 resulting from the converting and clipping stages 750 and  
14 768. It would be advantageous to eliminate or reduce these  
15 errors.

16 An example of a system which embodies the process shown  
17 in Figure 6, is shown in Figure 7(b). The system 705  
18 performs real-domain manipulation on coded data. The coded  
19 data 'C' 715 are input to the entropy decoder 725, which is  
20 defined for JPEG-coded data in the JPEG standard. The  
21 entropy decoder 725 decompresses the data into quantized DCT  
22 coefficients. These quantized coefficients are input to the  
23 inverse quantizer 735 and the output passed to the inverse  
24 transformer, which in this system is the two-dimensional 8x8  
25 inverse DCT-er 745. The resulting real-valued image data  
26 are rounded to integers and clipped 755 (e.g. to the range  
27 [0,255]) to produce integer-valued image data 759 in the  
28 allowed range.

1        If it is necessary to show the data before  
 2 manipulation, for example when the image manipulation is an  
 3 interactive process, the image can optionally be sent to a  
 4 display 763. The image is operated on by a manipulator 767  
 5 to produce some desired change. If the result of the  
 6 manipulation is non-integer data then the image data may be  
 7 passed to another integer converter and clipper 773. In  
 8 this way the image data 777 may again be displayed 763. The  
 9 new real-domain image data 777 are passed to the forward  
 10 DCT-er 781 and the resulting DCT coefficients are input to  
 11 the quantizer 785 to produce new quantized DCT coefficients  
 12 789. These coefficients 789 are then input to the entropy  
 13 encoder 793 to produce new coded data 'C1' 797 which are  
 14 different from the original coded data 'C' 715. Now the new  
 15 coded data 'C1' 797 incorporates not only the desired  
 16 changes made to the image by the image manipulator 767, but  
 17 also the errors resulting from the integer converter and  
 18 clippers 755 and 773.

19        Figure 8(a) shows the JPEG example luminance  
 20 quantization matrix 804 for 8x8 DCT luminance blocks. Figure  
 21 8(b) gives the JPEG example chrominance quantization matrix  
 22 814 for 8x8 DCT chrominance blocks. The smallest  
 23 quantization value in Figure 8(a) is 10. The smallest  
 24 quantization value in Figure 8(b) is 17. Since the maximum  
 25 possible error from rounding is 0.5 for each of 64 samples,  
 26 the largest error in the unquantized forward transform  
 27 coefficients from conversion to integers by rounding is 4  
 28 (shown in Figure 8(c)) for JPEG. For the quantization  
 29 matrices shown in Figures 8(a) and 8(b) this size error is  
 30 less than half of all of the values and will disappear

1 during quantization. However, for high quality applications  
2 such as high end printing or digital studio editing, the  
3 quantization matrix values are much smaller. In some cases,  
4 the DC (upper left corner) term is as small as 1 to preserve  
5 maximum quality. Then the rounding errors are significant.

6 The maximum possible error from truncating is just  
7 under 1 for each sample. This almost doubles the error in  
8 the unquantized forward transform coefficients. For the  
9 quantization matrix in Figure 8(a) eight quantization values  
10 are small enough for this error to potentially change the  
11 transform-coded data.

12 A numerical example showing the multi-generation  
13 problem is given in Figure 8(c). In this example the  
14 transform used is the 8x8 DCT as used in the JPEG still  
15 image compression standard. A set of transform-domain  
16 coefficients **822**, of which only one (the constant, or DC,  
17 term) is non-zero, are operated on by the inverse transform  
18 to produce an block of real-domain data **824**. In this case  
19 the data consist of 64 values which are all equal to 128.5.  
20 Note that the JPEG level shift of 128 for 8 bit data has  
21 been applied. The real-domain data are rounded to the  
22 nearest integers **826**, which in this case means that each  
23 value is rounded up to 129. The forward transform is then  
24 applied to produce new transform-domain coefficients **828**.  
25 It can be seen that the resulting new transform coefficients  
26 **828** are significantly different from the initial transform  
27 coefficients **822**. This is a highly undesirable result.

28 This example also applies to transform-coded data if  
29 the DC quantization value is set to 1, 2, or 4. Then the

1 transform coefficients **822** would be produced from  
2 transform-coded values of 4, 2, or 1 respectively. The  
3 quantization of the new transform coefficients **828** would  
4 change the resulting DC quantization values to 2, 4, or 8  
5 respectively.

6 Another numerical example showing the multi-generation  
7 problem is given in Figure 8(d). Again the transform used  
8 is the 8x8 DCT as used in the JPEG still image compression  
9 standard. A set of transform-domain coefficients **832**, of  
10 which only one (the constant, or DC, term) is non-zero, are  
11 operated on by the inverse transform to produce a block of  
12 real-domain data **834**. In this case the data consist of 64  
13 values which are all equal to 128.875. Note that the JPEG  
14 level shift of 128 for 8 bit data has been applied. The  
15 real-domain data are truncated to the nearest integers **836**,  
16 which in this case means that each value is reduced to 128.  
17 The forward transform is then applied to produce new  
18 transform-domain coefficients **838**. It can be seen that the  
19 resulting new transform coefficients **838** are significantly  
20 different from the initial transform coefficients **832**. This  
21 is a highly undesirable result.

22 Having demonstrated the errors caused by real-domain  
23 rounding or truncating when converting to integers, we now  
24 show how real-domain clipping can cause errors. Figure 8(e)  
25 shows an example of real-domain clipping **850**. This example  
26 uses the one-dimensional DCT to illustrate the problem.  
27 Figure 8(d) shows a bar chart **854** displaying one block of  
28 data consisting of eight samples. The data displayed has

1 only two frequency components: a constant, or DC, component  
2 which is indicated by the dotted line; and an alternating,  
3 or AC, component which gives an alternating wave pattern  
4 symmetrical about the dotted line. The magnitudes of these  
5 components, namely the respective DCT coefficients, are  
6 high-precision numbers. When quantization is performed,  
7 these DCT coefficients are rounded to the nearest  
8 quantization level. The data after transform-domain  
9 quantization are shown in the bar chart **858**. In the example  
10 shown, the DC coefficient has a small quantization value and  
11 so quantization does not change the DC level significantly.  
12 The AC coefficient shown has a large quantization value and  
13 so is changed significantly by quantization. This example  
14 shows the AC component almost doubling in magnitude due to  
15 quantization. These quantization values reflect, for  
16 example, those used when compressing chrominance image data.  
17 Thus the data represented after quantization have parts  
18 which have negative values. This shows how transform-domain  
19 data which, after inverse transforming, give real-domain  
20 negative values can be produced by original real-domain data  
21 which do not contain negative values.

22 Bar chart **862** shows the data produced from that in  
23 chart **858** after real-domain clipping. Those negative parts  
24 of the real data have been changed to 0. This results in  
25 the DC coefficient of the data increasing and hence leads to  
26 error being introduced. Because the quantization value for  
27 the DC coefficient is generally small, the error is large  
28 enough to cause a change in the quantized data as given in  
29 Equation (3).



1       To further illustrate the possibility of error  
2 introduced by real-domain clipping, a numerical example 870  
3 is shown in Figures 8(f) and 8(g). This example employs the  
4 system illustrated in Figure 5. This example uses the  
5 two-dimensional 8x8 DCT as used for transform coding of  
6 images to illustrate the problem described above. The  
7 initial quantized DCT coefficients are shown in matrix 874.  
8 All but two of the coefficients are 0; the two non-zero  
9 coefficients are the DC coefficient and one high-frequency  
10 coefficient. The coefficients, after dequantizing using the  
11 quantization matrix shown in Figure 8(a), are shown in  
12 matrix 878. When the inverse DCT is performed on these  
13 transform data and the level shift of 128 added, real data  
14 are produced as shown in matrix 882. The data shown in  
15 matrix 882 have already been rounded to integers but have  
16 not been clipped to an allowed range. It can be seen that  
17 these real data include several negative values. After  
18 clipping, the real data 882 produce clipped real data as  
19 shown in matrix 886. These data are identical to 882 except  
20 that each negative value has been replaced by 0. The  
21 forward DCT is then applied to the real-domain data to give  
22 new rounded transform data 890. It can be seen that the new  
23 transform data are significantly different from the previous  
24 transform data 878. When quantization is performed using  
25 the quantization matrix shown in Figure 8(a), new  
26 transform-coded data 894 are produced. The resulting  
27 changes in the transform data are large enough to produce  
28 changes in the transform-coded data after quantization.  
29 This is a highly undesirable result.

1        In many situations, the process of decoding,  
2 manipulation and re-encoding of data needs to be done  
3 multiple times. In these situations each iteration of this  
4 process is referred to as a "generation". The errors  
5 described above, caused by converting to integers and  
6 clipping to an allowed range in the real domain, accumulate  
7 as multiple iterations are performed and may result in  
8 significant degradation of the data. It is realized that the  
9 foregoing are only representative examples of errors  
10 introduced by rounding (or truncating) and/or clipping.  
11 Other examples having more or less error developed are  
12 possible.

13        The problem is usually even worse following multiple  
14 generations of decoding and re-encoding as shown in Figure  
15 9. Initial transform-coded data 'D0' 910 is dequantized and  
16 inverse transformed 920, converted to integers and clipped  
17 to an allowed range 930 to produce integer-valued  
18 real-domain data 940. The real-domain data 940 are passed  
19 to the forward transform and quantized 950 to give new  
20 transform-coded data 'D1' 960. This whole process is  
21 iterated several times, and after some number 'n' of  
22 iterations the final transform-coded data 'Dn' 970 is  
23 produced. Because of errors in each step the final data  
24 'Dn' 970 may be very different from the original data.

25        A case showing the problem significantly worsened due  
26 to multiple generations of real-domain manipulation of  
27 transform-coded data is shown in Figure 10. In addition to  
28 the steps shown in Figure 9, some form of manipulation 310  
29 is performed on the real-domain data, followed by converting  
30 to integers and clipping 320. After the forward transform

1 and quantization, the resulting quantized transform  
2 coefficients 1010 contain some error as in Figure 5. After  
3 'n' generations, the final transform quantized coefficients  
4 1020 may have quite large undesired errors.

## 5 DETAILED DESCRIPTION OF THE INVENTION

6 An example embodiment of a method for processing  
7 transform data with reduced error 1100 is illustrated in  
8 Figure 11(a). Transform data 'A' 110 are passed through an  
9 inverse transform 120 to produce high-precision real-domain  
10 data 130, as in Figure 1(a). If it is necessary to pass the  
11 real-domain data to an output device 160 which takes  
12 integer-valued input, or to generate integer-valued data  
13 before manipulation for any other reason, the steps of  
14 converting to integers and clipping to an allowed range 140  
15 is done before manipulation without affecting the high-  
16 precision real-domain data. The desired manipulation 1110  
17 of the real-domain data is performed using a method which  
18 accepts high-precision data as input and produces  
19 high-precision data 1120 as output. This manipulation  
20 method 1110 performs conceptually the same processing on the  
21 data as the manipulation on integers 310 described above in  
22 Figure 3, but operates instead on high-precision data. If  
23 it is necessary to pass the manipulated real-domain data to  
24 an output device 160 which takes integer-valued input, or to  
25 generate integer-valued data after manipulation for any  
26 other reason, the steps of converting to integers and  
27 clipping to an allowed range 140 are done after manipulation  
28 without affecting the high precision of the processed data.

1       An example embodiment of a system for processing  
2 transform data with reduced error 1105 in accordance with  
3 the present invention is illustrated in Figure 11(b).  
4 Transform data 'A' 115 are passed through an inverse  
5 transformer 125 to produce high-precision real-domain data  
6 135, as in Figure 1(b). If it is necessary to pass the  
7 real-domain data to an output device 165 which takes  
8 integer-valued input, or to generate integer-valued data  
9 before manipulation for any other reason, the integer  
10 converter and clipper 145 operates before manipulation  
11 without affecting the high-precision real-domain data. The  
12 manipulator 1115 operates on the real-domain data accepting  
13 high-precision data as input and producing high-precision  
14 data 1125 as output. This manipulator 1115 performs  
15 conceptually the same processing on the data as the  
16 manipulation on integers 310 described above in Figure 3,  
17 but operates instead on high-precision data. If it is  
18 necessary to pass the manipulated real-domain data to an  
19 output device 165 which takes integer-valued input, or to  
20 generate integer-valued data after manipulation for any  
21 other reason, the integer converter and clipper 145 operates  
22 after manipulation without affecting the high precision of  
23 the processed data.

24       An example of an embodiment of the present invention  
25 employing a method for performing inverse transform followed  
26 by forward transform steps 1200 is illustrated in Figure  
27 12(a). Transform data 'A' 110 are passed through an inverse  
28 transform 120 to produce high-precision real-domain data  
29 130, as in Figure 1(a). If it is necessary to pass the  
30 real-domain data to an output device 160 which takes

1 integer-valued input, or to generate integer-valued data for  
2 any other reason, the steps of converting to integers and  
3 clipping to an allowed range 140 are done without affecting  
4 the high-precision real-domain data. The high-precision  
5 data 130 are used as input to the forward transform 1210,  
6 which accepts real-valued data as input. The resulting  
7 transform data 'A3' 1220 are identical to the original  
8 transform data 'A' 110 which were the input to the inverse  
9 transform 120 if the forward transform 1210 is the inverse  
10 of the inverse transform since the errors from rounding and  
11 clipping are not present in the transform data 'A3'. The  
12 forward transform 1210 will produce different transform data  
13 'A3' 1220 when a different forward transform is used. This  
14 allows conversion between transforms without the errors from  
15 rounding and clipping being present in the forward transform  
16 input.

17 An example of an embodiment of the present invention  
18 employing a system with an inverse transformer followed by  
19 forward transformer 1205 is illustrated in Figure 12(b).  
20 Transform data 'A' 115 are passed through an inverse  
21 transformer 125 to produce high-precision real-domain data  
22 135, as in Figure 1(b). If it is necessary to pass the  
23 real-domain data to an output device 165 which takes  
24 integer-valued input, or to generate integer-valued data for  
25 any other reason, the integer converter and clipper 145  
26 operates without affecting the high-precision real-domain  
27 data 135. The high-precision data 135 are used as input to  
28 the forward transform 1215, which accepts real-valued data  
29 as input. The resulting transform data 'A3' 1225 are

1 identical to the original transform data 'A' 115 which were  
2 the input to the inverse transformer 125 if the forward  
3 transformer 1215 implements the inverse of the inverse  
4 transform since the errors from rounding and clipping are  
5 not present in the transform data 'A3'. The forward  
6 transformer 1215 will produce different transform data 'A3'  
7 1225 when a different forward transformer is used.

8 Figure 13(a) shows a method for performing real-domain  
9 manipulation of transform data with reduced error 1300. This  
10 method is formed by extending the method 1100 described in  
11 Figure 11(a). In this case, the high-precision data 1120 are  
12 passed as input to a forward transform 1210 which accepts  
13 high-precision data as input, to produce new transform data  
14 'A4' 1310 without rounding and/or clipping errors.

15 Figure 13(b) shows a system for performing real-domain  
16 manipulation of transform data with reduced error 1305. This  
17 method is formed by extending the system 1105 described in  
18 Figure 11(b). In this case, the high-precision data 1125 are  
19 passed as input to a forward transformer 1215 which accepts  
20 high-precision data as input, to produce new transform data  
21 'A4' 1315 without rounding and/or clipping errors.

22 A method for performing real-domain manipulation of  
23 transform-coded data with reduced error is illustrated in  
24 Figure 14(a). Figure 14(a) shows integer transform-coded  
25 data 'B' 210 are dequantized 220 and the output passed  
26 through an inverse transform 120 to produce high-precision  
27 real-domain data 130, as in Figure 2(a). If it is necessary  
28 to pass the real-domain data 130 to an output device 160  
29 which takes integer-valued input, or to generate

1 integer-valued data 150 before manipulation for any other  
2 reason, the steps of converting to integers and clipping to  
3 an allowed range 140 are done before manipulation without  
4 affecting the high-precision real-domain data 130. The  
5 desired manipulation 1110 of the real-domain data is then  
6 performed using a method which accepts high-precision data  
7 as input and produces high-precision data 1410 as output.  
8 This manipulation 1110 performs conceptually the same  
9 processing on the data as the manipulation on integers 310  
10 described above in Figure 3, but operates instead on  
11 high-precision data. If it is necessary to pass the  
12 manipulated real-domain data to an output device 160 which  
13 takes integer-valued input, or to generate integer-valued  
14 data after manipulation for any other reason, the steps of  
15 converting to integers and clipping to an allowed range 140  
16 are done after manipulation 1110 without affecting the high  
17 precision of the processed data 1410.

18 A system for performing real-domain manipulation of  
19 transform-coded data with reduced error is illustrated in  
20 Figure 14(b). Figure 14(b) shows integer transform-coded  
21 data 'B' 215 input to an inverse quantizer 225 and passed  
22 through an inverse transformer 125 to produce high-precision  
23 real-domain data 135, as in Figure 2(b). If it is necessary  
24 to pass the real-domain data 135 to an output device 165  
25 which takes integer-valued input, or to generate  
26 integer-valued data 155 before manipulation for any other  
27 reason, the integer converter and clipper 145 operates on  
28 the data before manipulation without affecting the  
29 high-precision real-domain data 135. The desired

1 manipulation of the real-domain data is then performed using  
2 a manipulator 1115 which accepts high-precision data as  
3 input and produces high-precision data 1415 as output. This  
4 manipulator 1115 performs conceptually the same processing  
5 on the data as the manipulation on integers 310 described  
6 above in Figure 3, but operates instead on high-precision  
7 data. If it is necessary to pass the manipulated  
8 real-domain data to an output device 165 which takes  
9 integer-valued input, or to generate integer-valued data  
10 after manipulation for any other reason, the integer  
11 converter and clipper 145 operates on the non-integer data  
12 1415 after manipulation 1115 without affecting the high  
13 precision of the processed data 1415.

14 An example embodiment of a method for real-domain  
15 conversion of transform-coded data 1500 is shown in Figure  
16 15(a). The high-precision data 130 are used as input to the  
17 forward transform 1210, which accepts real-valued data as  
18 input. The output of the forward transform 1210 is quantized  
19 1510. Depending upon the desired system implementation, the  
20 forward transform operation 1210 may employ a different  
21 transform than that used in the inverse transform operation  
22 120. For example, the inverse transform 120 may use the  
23 inverse DCT transform whereas the forward transform 1210 may  
24 use the Fourier transform. The resulting integer  
25 transform-coded data 'B2' 1520 are identical to the original  
26 integer transform-coded data 'B' 210 which were the input to  
27 the inverse quantize step 220 if the forward transform  
28 operation 1210 is the inverse of the inverse transform  
29 operation 120 and the quantization values used in the



1 inverse quantization step 220 and the quantization step 1510  
2 are identical. It is noted that the forward transform 1210  
3 will produce different integer transform-coded data 'B2'  
4 when a different forward transform is used. Similarly, use  
5 of different quantization values in the inverse quantization  
6 220 and quantization 1510 also produces different integer  
7 transform-coded data 1520. This method thus allows  
8 conversion between transforms and quantization matrices  
9 without the errors from rounding and clipping being present  
10 in the forward transform 1210 input 130.

11 The conversion between quantization matrices may be for  
12 coarser or finer quantization. For converting data from the  
13 JPEG international standard to the MPEG international  
14 standard, the quantization is likely to be coarser. The  
15 higher quality JPEG independent images are needed during the  
16 editing process. The coarser, more compressible, MPEG  
17 images are used to achieve the desired bandwidth objectives.  
18 On the other hand, in recompressing JPEG images after  
19 significant hand editing, the quantization is likely to be  
20 finer in order to preserve the changes.

21 An example embodiment of a system for real-domain  
22 conversion of transform-coded data 1505 in accordance with  
23 the present invention is shown in Figure 15(b). The  
24 high-precision data 135 are used as input to the forward  
25 transformer 1215, which accepts real-valued data as input.  
26 The output of the forward transformer 1215 is input to the  
27 quantizer 1515. Depending upon the desired system  
28 implementation, the forward transformer 1215 may produce a  
29 different transform than that used in the inverse

1 transformer 125. For example, the inverse transformer 125  
2 may use the inverse DCT transform whereas the forward  
3 transformer 1215 may use the Fourier transform. The  
4 resulting integer transform-coded data 'B2' 1525 are  
5 identical to the original integer transform-coded data 'B'  
6 215 which was the input to the inverse quantizer 225 if the  
7 forward transformer 1215 produces the inverse of the inverse  
8 transformer 125 and the quantization values used in the  
9 inverse quantizer 225 and the quantizer 1515 are identical..  
10 It is noted that the forward transformer 1215 will produce  
11 different integer transform-coded data 'B2' when a different  
12 forward transform is produced. Similarly, use of different  
13 quantization values in the inverse quantizer 225 and  
14 quantizer 1515 also produces different integer  
15 transform-coded data 1525. This system thus allows  
16 conversion between transforms and quantization matrices  
17 without the errors from rounding and clipping being present  
18 in the forward transformer 1215 input 135.

19 A method for performing real-domain manipulation of  
20 transform-coded data with reduced error 1600 is formed by  
21 extending the method 1400 described in Figure 14(a) as is  
22 illustrated in Figure 16(a). The high-precision data 1410  
23 are passed as input to a forward transform 1210 which  
24 accepts high-precision data as input. The output values from  
25 the forward transform are quantized 1510 to produce new  
26 transform-coded data 'B3' 1610.

27 A system for performing real-domain manipulation of  
28 transform-coded data with reduced error 1605 is formed by  
29 extending the method 1405 described in Figure 14(b) as is

1 illustrated in Figure 16(b). The high-precision data 1415  
2 are passed as input to a forward transformer 1215 which  
3 accepts high-precision data as input. The output values from  
4 the forward transformer are input to the quantizer 1515 to  
5 produce new transform-coded data 'B3' 1615.

6 An example embodiment of a method for real-domain  
7 manipulation of transform-coded data with reduced error 1700  
8 is shown in Figure 17(a). The chosen embodiment is a method  
9 for real-domain manipulation of coded images, which are  
10 transform-coded using the DCT. Coded data 'C' 710 are  
11 decoded by a lossless entropy decode step 720 to produce  
12 quantized DCT coefficients. These coefficients are  
13 dequantized 730 and passed through an inverse DCT 740 to  
14 produce high-precision real-domain data 1710. If it is  
15 necessary to pass the image before manipulation to a display  
16 device 758 which takes integer-valued input, or to produce  
17 integer-valued data 754 before manipulation for any other  
18 reason, the steps of converting to integers and clipping to  
19 an allowed range 750 are performed before manipulation 1720  
20 without affecting the high-precision real-domain image data  
21 1710. The desired manipulation 1720 of the image is then  
22 performed using a method which accepts high-precision data  
23 as input and produces high-precision data 1730 as output.  
24 If it is necessary to pass the manipulated image data to a  
25 display 758 which takes integer-valued input, or to generate  
26 integer-valued image data 1750 after manipulation for any  
27 other reason, the steps of converting to integers and  
28 clipping to an allowed range 1740 are performed after  
29 manipulation 1720 without affecting the high precision of

1 the processed image data 1730. The high-precision image  
2 data 1730 are passed as input to a forward DCT 1760 which  
3 accepts high-precision data as input. The output values  
4 from the forward transform 1760 are quantized 780 to produce  
5 new integer DCT coefficients 1770. These coefficients 1770  
6 are encoded by a lossless entropy encode step 788 to produce  
7 new coded data 'C2' 1780. If the forward and inverse  
8 transforms and the manipulation system are sufficiently  
9 accurate so that the error they introduce is less than half  
10 a quantization step, as described in Equation (3) given  
11 above, no error at all is introduced to the DCT  
12 coefficients.

13 An example invention embodiment of a system for  
14 real-domain manipulation of transform-coded data with  
15 reduced error 1705 is shown in Figure 17(b). The chosen  
16 embodiment is to implement a method for real-domain  
17 manipulation of coded images such as JPEG-coded images,  
18 which are transform-coded using the DCT. Coded data 'C' 715  
19 are decoded by a lossless entropy decoder 725 to produce  
20 quantized DCT coefficients. These coefficients are sent to  
21 a inverse quantizer 735 and then passed through an inverse  
22 DCT-er 745 to produce high-precision real-domain data 1715.  
23 If it is necessary to pass the image before manipulation to  
24 a display device 763 which takes integer-valued input, or to  
25 produce integer-valued data 759 before manipulation for any  
26 other reason, the integer converter and clipper 755 produces  
27 integer-valued data in the allowed range before manipulation  
28 1725 without affecting the high-precision real-domain image  
29 data 1715. The manipulator 1725 which performs the desired  
30 manipulation of the image accepts high-precision data as

1 input and produces high-precision data 1735 as output. If  
2 it is necessary to pass the manipulated image data to a  
3 display 763 which takes integer-valued input, or to generate  
4 integer-valued image data 1755 after manipulation for any  
5 other reason, the optional integer converter and clipper  
6 1745 produces integer-valued data 1755 after the operation  
7 of the manipulator 1725 without affecting the high precision  
8 of the processed image data 1735. The high-precision image  
9 data 1735 are passed as input to a forward DCT-er 1765 which  
10 accepts high-precision data as input. The output values  
11 from the forward DCT-er 1765 are sent to the quantizer 785  
12 to produce new integer DCT coefficients 1775. These  
13 coefficients 1775 are encoded by a lossless entropy encoder  
14 793 to produce new coded data 'C2' 1785. If the forward and  
15 inverse transforms and the manipulation system are  
16 sufficiently accurate so that the error they introduce for  
17 each coefficient is less than half a quantization step, as  
18 described in Equation (3) given above, no additional error  
19 is introduced to the DCT coefficients.

20 A method for performing real-domain manipulations of  
21 transform-coded data with reduced error in multiple steps  
22 1800, alternating the manipulation steps with forward  
23 transforming and quantizing steps and inverse transform and  
24 quantizing steps, is illustrated in Figure 18(a). In general  
25 each manipulation may perform another operation on the data.  
26 For example for digital studio editing, the first  
27 manipulation might color correct the image. The second  
28 manipulation might merge the color corrected image with a  
29 background using the chroma-keying method. The third  
30 manipulation might add highlights to the image. The fourth

1 manipulation might crop the image to convert from the 16:9  
2 width to height aspect ratio of movies to the 4:3 aspect  
3 ratio of television. For the printing of images the first  
4 manipulation might rotate the image 90 degrees to orient the  
5 image with the printing direction. The second manipulation  
6 might merge several independent images into one composite  
7 image. A third manipulation might do a color conversion.

8       As shown in Figure 18(a) transform-coded data 'D0' 910  
9 are dequantized and passed through an inverse transform 920  
10 to produce high-precision real-domain data 1810. If it is  
11 necessary to produce integer-valued data for any reason, the  
12 high-precision data 1810 may be converted to integers and  
13 clipped to an allowed range 1820 without affecting the high  
14 precision of the real-domain data 1810. The desired  
15 manipulation 1110 of the real-domain data is then performed  
16 using a method which accepts high-precision data 1810 as  
17 input and produces high-precision data 1840 as output. If  
18 it is desired to produce an integer-valued of this output  
19 data, the high-precision data 1810 may be converted to  
20 integers and clipped to an allowed range 1830 without  
21 affecting the high precision of the output data. The  
22 high-precision output data are passed as input to a forward  
23 transformer and quantizer 1850 to produce new  
24 transform-coded data 'F1' 1860. The process of inverse  
25 quantizing and inverse transforming, manipulation and  
26 forward transforming and quantizing may be repeated multiple  
27 times with the manipulation 1870 being different upon each  
28 iteration. After multiple steps, final transform-coded data  
29 'Fn' 1880 are produced with rounding and/or clipping errors  
30 reduced or eliminated. Outputs resulting from any of the

1 convert to integer and clip steps may be sent to an output  
2 device 1890 with or without a multiplexor.

3 An example invention embodiment of a system for  
4 performing real-domain manipulations of transform-coded data  
5 with reduced error in multiple stages 1805, alternating the  
6 operation of a manipulator with the operation of a forward  
7 transformer and quantizer and the operation of an inverse  
8 quantizer and inverse transformer, is illustrated in Figure  
9 18(b). Transform-coded data 'D0' 1815 are fed to an inverse  
10 quantizer and inverse transformer 1819 to produce  
11 high-precision real-domain data 1823. If it is necessary to  
12 produce integer-valued data for any reason, the  
13 high-precision data 1823 may be operated on by the integer  
14 converter and clipper 1827 without affecting the high  
15 precision of the real-domain data 1823. The manipulator  
16 1115 then operates on the real-domain data 1823 to produce  
17 the desired manipulation and produces high-precision data  
18 1845 as output. If it is desired to produce integer-values  
19 of this output data, the high-precision data 1845 may be  
20 input to an integer converter and clipper 1835 without  
21 affecting the high precision of the output data. The  
22 high-precision output data are passed as input to a forward  
23 transformer and quantizer 1855 to produce new  
24 transform-coded data 'F1' 1865. The steps of inverse  
25 quantizing and inverse transforming, manipulation and  
26 forward transforming and quantizing may be repeated multiple  
27 times with the manipulator 1875 being different upon each  
28 iteration. After multiple iterations, final transform-coded  
29 data 'Fn' 1885 are produced with real-domain rounding and/or  
30 clipping errors reduced or eliminated. In a particular

1 embodiment the output from any or all of the integer  
2 converter and clipper modules is fed to the output device  
3 **1895**. For coded image data the output device may be a  
4 display or television set. For coded audio data the output  
5 device may be a player and/or recorder.

6 A numerical example showing how the present invention  
7 solves one aspect of the multi-generation problem is given  
8 in Figure 19(a). A set of transform-domain coefficients  
9 **822**, of which only one (the constant, or DC, term) is  
10 non-zero, are operated on by the inverse transform to  
11 produce an block of real-domain data **824**. In this case the  
12 data consist of 64 values which are all equal to 128.5.  
13 Note that the JPEG level shift of 128 for 8 bit data has  
14 been applied. The forward transform is then applied to  
15 produce new transform-domain coefficients **1910**. It can be  
16 seen that the new transform coefficients **1910** are identical  
17 to the initial transform coefficients **822**. This is due to  
18 the rounding error not being present in the data sent to the  
19 forward DCT.

20 Another numerical example showing how the present  
21 invention solves another aspect of the multi-generation  
22 problem is given in Figure 19(b). A set of transform-domain  
23 coefficients **832**, of which only one (the constant, or DC,  
24 term) is non-zero, are operated on by the inverse transform  
25 to produce an block of real-domain data **834**. In this case  
26 the data consist of 64 values which are all equal to  
27 128.875. Note that the JPEG level shift of 128 for 8 bit  
28 data has been applied. The forward transform is then applied  
29 to produce new transform-domain coefficients **1938**. It can  
30 be seen that the new transform coefficients **1938** are



1 identical to the initial transform coefficients **832**. This  
2 is due to the truncation error not being present in the data  
3 sent to the forward DCT.'

4       Having demonstrated how using the high-precision  
5 numbers removes the errors caused by real-domain rounding or  
6 truncating, we now show how real-domain clipping errors are  
7 also avoided. The same numerical starting point and first  
8 three steps used in Figure 8(f) are shown in Figure 19(c).  
9 The initial quantized DCT coefficients are shown in matrix  
10 **874**. All but two of the coefficients are 0; the two  
11 non-zero coefficients are the DC coefficient and one  
12 high-frequency coefficient. The coefficients after  
13 dequantizing are shown in matrix **878**. The quantization  
14 matrix used is shown in Figure 8(a). When the inverse DCT  
15 is performed on these transform data, real data are produced  
16 as shown in matrix **882**. The data shown in matrix **882** have  
17 already been rounded to integers but have not been clipped  
18 to an allowed range.

19       Figure 19(d) shows the results of the forward DCT  
20 applied to the real-domain data to give new rounded  
21 transform data **1944**. When quantization is performed, new  
22 transform-coded data **1948** are produced. In this example,  
23 the changes in the transform data are not large enough to  
24 produce changes in the transform-coded data after  
25 quantization.

26       Examples of the manipulation between generations  
27 include merging two or more transform-coded data sets. For  
28 transform-coded image data sets, the merging may be needed  
29 because multiple small images need to be collected into one

1 bigger picture. Fan-folded advertising brochures typically  
2 are composed of multiple individual pictures. Today's  
3 highest end laser printers print more than one page at a  
4 time. In such cases, the images generally do not overlap,  
5 but may not have the same quantization, positioning relative  
6 to the reference grid such as the 8x8 block structure for  
7 JPEG DCTs, or orientation. By composing the final picture  
8 in the real domain, standard processes can be used for each  
9 subimage. Then the composite image can be re-compressed for  
10 eventual decompression for on-the-fly printing.

11 Similarly, digital editing can include many special  
12 effects requiring several independent manipulations  
13 performed serially. Digital movies often use the  
14 fade-in/fade-out special effect to perform a smooth  
15 transition between two key scenes. Such special effects may  
16 follow independent processing of each scene. Thus, multiple  
17 generations of decompression and recompression are often  
18 needed in the editing to produce the composite of the  
19 special effects.

20 Chroma-keying involves two independent video data  
21 streams. In one video stream the background has been  
22 captured. In the other video stream the foreground, often  
23 composed of action involving live actors, has been filmed  
24 against a blank single color such as a deep blue or black  
25 background. Then the blank pixels in the foreground image  
26 are replaced with pixels from the background video. Since  
27 the pixels are being mixed at a single-pixel level, the  
28 images need to be combined in the real domain. The errors  
29 introduced by converting to integers and clipping are highly  
30 undesirable for such digital studio applications. These

1 errors are reduced or eliminated by implementing the present  
2 invention.

3 Another application example for use of the present  
4 invention is in the high-end digital graphics market which  
5 uses digital images with sometimes more than 100 megapixels.  
6 Glossy advertising brochures and the large photographic  
7 trade show booth backdrops are just two examples of the use  
8 of such high quality digital imagery. High-quality lossy  
9 JPEG compression are sometimes used to keep the transmission  
10 and storage costs down. As such images are decompressed and  
11 recompressed to allow changes and modifications such as  
12 adding highlights, correcting colors, adding or changing  
13 text and image cropping, unintentional changes are a problem  
14 that is solved with the use of the concepts of the present  
15 invention.

16 The above examples for the concepts of the present  
17 invention are usual for image and video transform data. The  
18 wide use of the Internet has shown the value of JPEG and  
19 MPEG compressed image data. When JPEG images are to be  
20 printed, then manipulations such as a change of scale or a  
21 change of orientation may be required. In addition, a  
22 transformation to another color space followed by  
23 recompression will allow the print-ready versions of the  
24 image to be stored. Use of the present invention overcomes  
25 the problem inherent in propagating the errors from the  
26 rounding and clipping.

27 Audio coded data also needs to be decompressed, mixed  
28 with special sound effects, merged with other audio data,  
29 edited and processed in the real domain with reduced errors.  
30 Similar implementations are performed for other industrial,

1 commercial, and military applications of digital processing  
2 employing a transform and an inverse transform of data  
3 representing a phenomenon when the data is stored in the  
4 transform domain. These are thus other representative  
5 applications wherein use of the present invention is highly  
6 advantageous.

7       It is further noted that this invention may also be  
8 provided as an apparatus or a computer product. For  
9 example, it may be implemented as an article of manufacture  
10 comprising a computer usable medium having computer readable  
11 program code means embodied therein for causing a computer  
12 to perform the methods of the present invention.

13       It is noted that although the description of the  
14 invention is made for particular arrangements of steps, the  
15 intent and concept of the present invention are suitable and  
16 applicable to other arrangements. It will be clear to those  
17 skilled in the art that other modifications to the disclosed  
18 embodiments can be effected without departing from the  
19 spirit and scope of the invention.